natural

Version 4.1.2 for Mainframes | Terminal Commands

SOFTWARE AG

This document applies to Natural Version 4.1.2 for Mainframes and to all subsequent releases.

Specifications contained herein are subject to change and these changes will be reported in subsequent release notes or new editions.

Table of Contents

Terminal Commands - Overview

This documentation describes the Natural terminal commands. The following topics are covered:

- Introduction to Terminal Commands
- Terminal Commands Grouped by Function
- Key Assignments

Introduction to Terminal Commands

The following topics are covered:

- What are Terminal Commands
- Issuing Terminal Commands
- Using Terminal Commands in Programs
- Terms Used in the Terminal Command Descriptions
- %? - Help for Terminal Commands - Mainframes only

What are Terminal Commands?

Terminal commands generally perform some terminal-related activities. The first character of a terminal command is the terminal command control character. It identifies the command as a terminal command. By default, the control character is "%".

Changing the Terminal Command Control Character

You can define another special character as control character; this is done with the session parameter CF.

When the control character is changed, all terminal commands which have been assigned to function keys will be adjusted accordingly.

Issuing Terminal Commands

Terminal commands can be used in a Natural runtime environment. The following rules apply:

- You can enter a terminal command in any unprotected field on the screen (including the message line if it is unprotected).
- A terminal command must be entered in one field, except the control character, which may be entered in a preceding field.
- It is recommended to enter a blank after a terminal command, or delete the remaining content of the field in which you enter the command. Otherwise Natural might misinterpret the field content as part of the terminal command.
- If you enter multiple terminal commands on the same screen, only the first command encountered will be executed, and all others will be ignored.
- Terminal commands which have been entered incorrectly are ignored, but you will not receive a corresponding error message.
- If you enter a terminal command and at the same time data in other fields on the screen, only the terminal command will be executed; the data will not be processed.

Using Terminal Commands in Programs

Terminal commands may also be issued from within a program by using the SET CONTROL statement. When a terminal command is specified with a SET CONTROL statement, the control character is omitted.

Terms Used in the Terminal Command Descriptions

In the descriptions of several terminal commands, the terms "screen" and "window" are used with the following meanings:

Screen	Depending on the operating system under which Natural is running, "screen" refers either to the entire terminal screen as such, or to the operating-system window in which the Natural session is running, or to the Natural main output window. However, for the sake of convenience, the term "screen" is used in all these instances.
Window	Always refers to the Natural window (as explained with the terminal command %W).

%? - Help for Terminal Commands - Mainframes only

%?

This terminal command displays help information on the Natural terminal commands.

Terminal Command List

The following table provides an overview of the terminal commands and the functions they perform.

Command	Function
%	Continuation indicator for INPUT statements in batch programs.
%%, **%.**	Interrupt current Natural operation.
%*	Disable display of input characters. In batch mode, suppress printing of next input record read.
%.P	Delete first entry from stack.
%.S	Read stack entry without deleting it.
%/	Forces end-of-file condition when using INPUT statement in batch mode.
%?	Invoke help information for terminal commands.
%+, **%-**	Enable/disable use of Natural Connection.
%<TECH	Display technical information.
%<TEST	Invoke the Natural Debugger.
%=	Assign colors to fields.
%A	Play back a recording.
%B	Activate/deactivate a recording.
%B=	Specify library for a recording.
%C	Copy current screen into Natural source area.
%CC, **%CS**	Copy data to the stack or into the system variable *COM.
%D	Activate keyword/delimiter mode.
%D=	Control of outlining.
%DUE	Dump for specific error.
%E	Display screen captures.
%E=	Activate/deactivate error processing.
%F	Activate forms/screen mode.
%F=	Frame characters for window.
%FM	Activate/deactivate edit mask free mode.
%G	Set playback mode for a recording.
%H	Produce hardcopy output.
%I	Capture current screen.
%J	Invoke helproutine.
%KN, **%KO**, **%KS**	Function-key logic for Siemens terminals.
%K*nn*, **%KP***n*	Simulate PF- and PA-keys.
%L	Disable lower- to upper-case translation.

Command	Function
%L=	Set language code.
%M	Control of message line.
%MSGSF	Display system error messages in full.
%N	Activate non-conversational mode.
%O	Deactivate screen capturing.
%P	Capture screen sequence.
%P=	CALL options.
%Q	Suppress map printing in batch mode.
%QO	Suppress pseudo-conversational output.
%QS	Simultaneous output of multiple screens.
%R	Repeat INPUT statement.
%RM	Protection of light pen sensitive fields.
%RN	Suppress compression of screen data.
%RO	Activate/deactivate screen optimization.
%S	Resume screen capturing.
%T	Position cursor on top of screen.
%T*ll/cc*	Set cursor position.
%T+ and **%T-**	Position cursor on protected field.
%T*	Position cursor outside window.
%T=	Activate converter routine for specific device type.
%TRE	Activate/deactivate external trace.
%TRI	Activate/deactivate internal trace.
%U	Enable lower- to upper-case translation.
%V	Control of print mode.
%W	Natural window handling.
%X	Control of statistics line/infoline.
%Y	Control of PF-key lines.
%Z	Clear source area.

Terminal Commands Grouped by Function

The following table provides an overview of the terminal commands grouped by functions.

- Case Translation
- Copy/Clear
- Language, Messages, Error Processing
- Screens, Terminal and Window Processing
- Colors, Outlining
- INPUT Statement, Stack
- Session Recording and Screen Capturing
- Statistics and Trace
- Miscellaneous
- Key Assignments

Case Translation	
%L	Disable lower- to upper-case translation.
%U	Enable lower- to upper-case translation.
Copy/Clear	
%C	Copy current screen into Natural source area.
%CC	Copy data into system variable *COM.
%CS	Copy data to the stack.
%Z	Clear source area.
Language, Messages, Error Processing	
%E=	Activate/deactivate error processing.
%L=	Set language code.
%M	Control of message line.
%MSGSF	Display system error messages in full.
Screens, Terminal and Window Processing	
%F=	Frame characters for window.
%K	Simulate PF- and PA-keys.
%K*nn*, **%KP***n*	Simulate PF- and PA-keys.
%KN, **%KO**, **%KS**	Function-key logic for Siemens terminals.
%N	Activate non-conversational mode.
%QS	Simultaneous output of multiple screens.
%RM	Protection of light pen sensitive fields.
%RN	Suppress compression of screen data.
%RO	Activate/deactivate screen optimization.
%T and **%T***ll*/*cc*	Set cursor position.
%T*	Position cursor outside window.
%T+ and **%T-**	Position cursor on protected field.
%W	Natural window handling.
%Y	Control of PF-key lines.
%*	Disable display of input characters. In batch mode, suppress printing of next input record read.
Colors, Outlining	
%D=	Control of outlining.
%=	Assign colors to fields.
INPUT Statement, Stack	
%	Continuation indicator for INPUT statements in batch programs.
%D	Activate keyword/delimiter mode.
%F	Activate forms/screen mode.

Case Translation	
%L	Disable lower- to upper-case translation.
%U	Enable lower- to upper-case translation.
Copy/Clear	
%C	Copy current screen into Natural source area.
%CC	Copy data into system variable *COM.
%CS	Copy data to the stack.
%Z	Clear source area.
Language, Messages, Error Processing	
%E=	Activate/deactivate error processing.
%L=	Set language code.
%M	Control of message line.
%MSGSF	Display system error messages in full.
Screens, Terminal and Window Processing	
%F=	Frame characters for window.
%K	Simulate PF- and PA-keys.
%K*nn*, **%KP***n*	Simulate PF- and PA-keys.
%KN, **%KO**, **%KS**	Function-key logic for Siemens terminals.
%N	Activate non-conversational mode.
%QS	Simultaneous output of multiple screens.
%RM	Protection of light pen sensitive fields.
%RN	Suppress compression of screen data.
%RO	Activate/deactivate screen optimization.
%T and **%T***ll/cc*	Set cursor position.
%T*	Position cursor outside window.
%T+ and **%T-**	Position cursor on protected field.
%W	Natural window handling.
%Y	Control of PF-key lines.
%*	Disable display of input characters. In batch mode, suppress printing of next input record read.
Colors, Outlining	
%D=	Control of outlining.
%=	Assign colors to fields.
INPUT Statement, Stack	
%FM	Activate/deactivate edit mask free mode.
%R	Repeat INPUT statement.
%/	Force end-of-file for INPUT in batch mode.

Case Translation	
%L	Disable lower- to upper-case translation.
%U	Enable lower- to upper-case translation.
Copy/Clear	
%C	Copy current screen into Natural source area.
%CC	Copy data into system variable *COM.
%CS	Copy data to the stack.
%Z	Clear source area.
Language, Messages, Error Processing	
%E=	Activate/deactivate error processing.
%L=	Set language code.
%M	Control of message line.
%MSGSF	Display system error messages in full.
Screens, Terminal and Window Processing	
%F=	Frame characters for window.
%K	Simulate PF- and PA-keys.
%K*nn*, **%KP***n*	Simulate PF- and PA-keys.
%KN, **%KO**, **%KS**	Function-key logic for Siemens terminals.
%N	Activate non-conversational mode.
%QS	Simultaneous output of multiple screens.
%RM	Protection of light pen sensitive fields.
%RN	Suppress compression of screen data.
%RO	Activate/deactivate screen optimization.
%T and **%T***ll*/*cc*	Set cursor position.
%T*	Position cursor outside window.
%T+ and **%T-**	Position cursor on protected field.
%W	Natural window handling.
%Y	Control of PF-key lines.
%*	Disable display of input characters. In batch mode, suppress printing of next input record read.
Colors, Outlining	
%D=	Control of outlining.
%=	Assign colors to fields.
INPUT Statement, Stack	
%.P	Delete first entry from Natural stack.
%.S	Read stack entry without deleting it.
Session Recording and Screen Capturing	

Case Translation	
%L	Disable lower- to upper-case translation.
%U	Enable lower- to upper-case translation.
Copy/Clear	
%C	Copy current screen into Natural source area.
%CC	Copy data into system variable *COM.
%CS	Copy data to the stack.
%Z	Clear source area.
Language, Messages, Error Processing	
%E=	Activate/deactivate error processing.
%L=	Set language code.
%M	Control of message line.
%MSGSF	Display system error messages in full.
Screens, Terminal and Window Processing	
%F=	Frame characters for window.
%K	Simulate PF- and PA-keys.
%K*nn*, **%KP***n*	Simulate PF- and PA-keys.
%KN, **%KO**, **%KS**	Function-key logic for Siemens terminals.
%N	Activate non-conversational mode.
%QS	Simultaneous output of multiple screens.
%RM	Protection of light pen sensitive fields.
%RN	Suppress compression of screen data.
%RO	Activate/deactivate screen optimization.
%T and **%T***ll/cc*	Set cursor position.
%T*	Position cursor outside window.
%T+ and **%T-**	Position cursor on protected field.
%W	Natural window handling.
%Y	Control of PF-key lines.
%*	Disable display of input characters. In batch mode, suppress printing of next input record read.
Colors, Outlining	
%D=	Control of outlining.
%=	Assign colors to fields.
INPUT Statement, Stack	
%A	Play back a recording.
%B	Activate/deactivate a recording.
%B=	Specify library for a recording.

Case Translation	
%L	Disable lower- to upper-case translation.
%U	Enable lower- to upper-case translation.
Copy/Clear	
%C	Copy current screen into Natural source area.
%CC	Copy data into system variable *COM.
%CS	Copy data to the stack.
%Z	Clear source area.
Language, Messages, Error Processing	
%E=	Activate/deactivate error processing.
%L=	Set language code.
%M	Control of message line.
%MSGSF	Display system error messages in full.
Screens, Terminal and Window Processing	
%F=	Frame characters for window.
%K	Simulate PF- and PA-keys.
%K*nn*, **%KP***n*	Simulate PF- and PA-keys.
%KN, **%KO**, **%KS**	Function-key logic for Siemens terminals.
%N	Activate non-conversational mode.
%QS	Simultaneous output of multiple screens.
%RM	Protection of light pen sensitive fields.
%RN	Suppress compression of screen data.
%RO	Activate/deactivate screen optimization.
%T and **%T***ll/cc*	Set cursor position.
%T*	Position cursor outside window.
%T+ and **%T-**	Position cursor on protected field.
%W	Natural window handling.
%Y	Control of PF-key lines.
%*	Disable display of input characters. In batch mode, suppress printing of next input record read.
Colors, Outlining	
%D=	Control of outlining.
%=	Assign colors to fields.
INPUT Statement, Stack	
%E	Display screen captures.
%G	Set playback mode for a recording.
%I	Capture current screen.

Case Translation	
%L	Disable lower- to upper-case translation.
%U	Enable lower- to upper-case translation.
Copy/Clear	
%C	Copy current screen into Natural source area.
%CC	Copy data into system variable *COM.
%CS	Copy data to the stack.
%Z	Clear source area.
Language, Messages, Error Processing	
%E=	Activate/deactivate error processing.
%L=	Set language code.
%M	Control of message line.
%MSGSF	Display system error messages in full.
Screens, Terminal and Window Processing	
%F=	Frame characters for window.
%K	Simulate PF- and PA-keys.
%K*nn*, **%KP***n*	Simulate PF- and PA-keys.
%KN, **%KO**, **%KS**	Function-key logic for Siemens terminals.
%N	Activate non-conversational mode.
%QS	Simultaneous output of multiple screens.
%RM	Protection of light pen sensitive fields.
%RN	Suppress compression of screen data.
%RO	Activate/deactivate screen optimization.
%T and **%T***ll/cc*	Set cursor position.
%T*	Position cursor outside window.
%T+ and **%T-**	Position cursor on protected field.
%W	Natural window handling.
%Y	Control of PF-key lines.
%*	Disable display of input characters. In batch mode, suppress printing of next input record read.
Colors, Outlining	
%D=	Control of outlining.
%=	Assign colors to fields.
INPUT Statement, Stack	
%O	Deactivate screen capturing.
%P	Capture screen sequence.
%S	Resume screen capturing.

Case Translation	
%L	Disable lower- to upper-case translation.
%U	Enable lower- to upper-case translation.
Copy/Clear	
%C	Copy current screen into Natural source area.
%CC	Copy data into system variable *COM.
%CS	Copy data to the stack.
%Z	Clear source area.
Language, Messages, Error Processing	
%E=	Activate/deactivate error processing.
%L=	Set language code.
%M	Control of message line.
%MSGSF	Display system error messages in full.
Screens, Terminal and Window Processing	
%F=	Frame characters for window.
%K	Simulate PF- and PA-keys.
%K*nn*, **%KP***n*	Simulate PF- and PA-keys.
%KN, **%KO**, **%KS**	Function-key logic for Siemens terminals.
%N	Activate non-conversational mode.
%QS	Simultaneous output of multiple screens.
%RM	Protection of light pen sensitive fields.
%RN	Suppress compression of screen data.
%RO	Activate/deactivate screen optimization.
%T and **%T***ll*/*cc*	Set cursor position.
%T*	Position cursor outside window.
%T+ and **%T-**	Position cursor on protected field.
%W	Natural window handling.
%Y	Control of PF-key lines.
%*	Disable display of input characters. In batch mode, suppress printing of next input record read.
Colors, Outlining	
%D=	Control of outlining.
%=	Assign colors to fields.
INPUT Statement, Stack	
Statistics and Trace	
%TRE	Activate/deactivate external trace.
%TRI	Activate/deactivate internal trace.

Case Translation	
%L	Disable lower- to upper-case translation.
%U	Enable lower- to upper-case translation.
Copy/Clear	
%C	Copy current screen into Natural source area.
%CC	Copy data into system variable *COM.
%CS	Copy data to the stack.
%Z	Clear source area.
Language, Messages, Error Processing	
%E=	Activate/deactivate error processing.
%L=	Set language code.
%M	Control of message line.
%MSGSF	Display system error messages in full.
Screens, Terminal and Window Processing	
%F=	Frame characters for window.
%K	Simulate PF- and PA-keys.
%K*nn*, **%KP***n*	Simulate PF- and PA-keys.
%KN, **%KO**, **%KS**	Function-key logic for Siemens terminals.
%N	Activate non-conversational mode.
%QS	Simultaneous output of multiple screens.
%RM	Protection of light pen sensitive fields.
%RN	Suppress compression of screen data.
%RO	Activate/deactivate screen optimization.
%T and **%T***ll/cc*	Set cursor position.
%T*	Position cursor outside window.
%T+ and **%T-**	Position cursor on protected field.
%W	Natural window handling.
%Y	Control of PF-key lines.
%*	Disable display of input characters. In batch mode, suppress printing of next input record read.
Colors, Outlining	
%D=	Control of outlining.
%=	Assign colors to fields.
INPUT Statement, Stack	
%X	Control of statistics line/infoline.
%<TECH	Display technical information.
%<TEST	Invoke the Natural Debugger.

Case Translation	
%L	Disable lower- to upper-case translation.
%U	Enable lower- to upper-case translation.
Copy/Clear	
%C	Copy current screen into Natural source area.
%CC	Copy data into system variable *COM.
%CS	Copy data to the stack.
%Z	Clear source area.
Language, Messages, Error Processing	
%E=	Activate/deactivate error processing.
%L=	Set language code.
%M	Control of message line.
%MSGSF	Display system error messages in full.
Screens, Terminal and Window Processing	
%F=	Frame characters for window.
%K	Simulate PF- and PA-keys.
%K*nn*, **%KP***n*	Simulate PF- and PA-keys.
%KN, **%KO**, **%KS**	Function-key logic for Siemens terminals.
%N	Activate non-conversational mode.
%QS	Simultaneous output of multiple screens.
%RM	Protection of light pen sensitive fields.
%RN	Suppress compression of screen data.
%RO	Activate/deactivate screen optimization.
%T and **%T***ll/cc*	Set cursor position.
%T*	Position cursor outside window.
%T+ and **%T-**	Position cursor on protected field.
%W	Natural window handling.
%Y	Control of PF-key lines.
%*	Disable display of input characters. In batch mode, suppress printing of next input record read.
Colors, Outlining	
%D=	Control of outlining.
%=	Assign colors to fields.
INPUT Statement, Stack	
Miscellaneous	
%H	Produce hardcopy output.
%J	Invoke helproutine.

Case Translation	
%L	Disable lower- to upper-case translation.
%U	Enable lower- to upper-case translation.
Copy/Clear	
%C	Copy current screen into Natural source area.
%CC	Copy data into system variable *COM.
%CS	Copy data to the stack.
%Z	Clear source area.
Language, Messages, Error Processing	
%E=	Activate/deactivate error processing.
%L=	Set language code.
%M	Control of message line.
%MSGSF	Display system error messages in full.
Screens, Terminal and Window Processing	
%F=	Frame characters for window.
%K	Simulate PF- and PA-keys.
%K*nn*, **%KP***n*	Simulate PF- and PA-keys.
%KN, **%KO**, **%KS**	Function-key logic for Siemens terminals.
%N	Activate non-conversational mode.
%QS	Simultaneous output of multiple screens.
%RM	Protection of light pen sensitive fields.
%RN	Suppress compression of screen data.
%RO	Activate/deactivate screen optimization.
%T and **%T***ll/cc*	Set cursor position.
%T*	Position cursor outside window.
%T+ and **%T-**	Position cursor on protected field.
%W	Natural window handling.
%Y	Control of PF-key lines.
%*	Disable display of input characters. In batch mode, suppress printing of next input record read.
Colors, Outlining	
%D=	Control of outlining.
%=	Assign colors to fields.
INPUT Statement, Stack	
%P=	CALL options.
%Q	Suppress map printing in batch mode.
%QO	Suppress pseudo-conversational output.

Case Translation	
%L	Disable lower- to upper-case translation.
%U	Enable lower- to upper-case translation.
Copy/Clear	
%C	Copy current screen into Natural source area.
%CC	Copy data into system variable *COM.
%CS	Copy data to the stack.
%Z	Clear source area.
Language, Messages, Error Processing	
%E=	Activate/deactivate error processing.
%L=	Set language code.
%M	Control of message line.
%MSGSF	Display system error messages in full.
Screens, Terminal and Window Processing	
%F=	Frame characters for window.
%K	Simulate PF- and PA-keys.
%K*nn*, **%KP***n*	Simulate PF- and PA-keys.
%KN, **%KO**, **%KS**	Function-key logic for Siemens terminals.
%N	Activate non-conversational mode.
%QS	Simultaneous output of multiple screens.
%RM	Protection of light pen sensitive fields.
%RN	Suppress compression of screen data.
%RO	Activate/deactivate screen optimization.
%T and **%T***ll*/*cc*	Set cursor position.
%T*	Position cursor outside window.
%T+ and **%T-**	Position cursor on protected field.
%W	Natural window handling.
%Y	Control of PF-key lines.
%*	Disable display of input characters. In batch mode, suppress printing of next input record read.
Colors, Outlining	
%D=	Control of outlining.
%=	Assign colors to fields.
INPUT Statement, Stack	
%T=	Activate converter routine for specific device type.
%V	Control of print mode.
%?	Invoke help information for terminal commands.

Case Translation	
%L	Disable lower- to upper-case translation.
%U	Enable lower- to upper-case translation.
Copy/Clear	
%C	Copy current screen into Natural source area.
%CC	Copy data into system variable *COM.
%CS	Copy data to the stack.
%Z	Clear source area.
Language, Messages, Error Processing	
%E=	Activate/deactivate error processing.
%L=	Set language code.
%M	Control of message line.
%MSGSF	Display system error messages in full.
Screens, Terminal and Window Processing	
%F=	Frame characters for window.
%K	Simulate PF- and PA-keys.
%K*nn*, **%KP***n*	Simulate PF- and PA-keys.
%KN, **%KO**, **%KS**	Function-key logic for Siemens terminals.
%N	Activate non-conversational mode.
%QS	Simultaneous output of multiple screens.
%RM	Protection of light pen sensitive fields.
%RN	Suppress compression of screen data.
%RO	Activate/deactivate screen optimization.
%T and **%T***ll*/*cc*	Set cursor position.
%T*	Position cursor outside window.
%T+ and **%T-**	Position cursor on protected field.
%W	Natural window handling.
%Y	Control of PF-key lines.
%*	Disable display of input characters. In batch mode, suppress printing of next input record read.
Colors, Outlining	
%D=	Control of outlining.
%=	Assign colors to fields.
INPUT Statement, Stack	
%% and **%.**	Interrupt current Natural operation.
%+ and **%-**	Enable/disable use of Natural Connection.
Key Assignments	

Case Translation	
%L	Disable lower- to upper-case translation.
%U	Enable lower- to upper-case translation.
Copy/Clear	
%C	Copy current screen into Natural source area.
%CC	Copy data into system variable *COM.
%CS	Copy data to the stack.
%Z	Clear source area.
Language, Messages, Error Processing	
%E=	Activate/deactivate error processing.
%L=	Set language code.
%M	Control of message line.
%MSGSF	Display system error messages in full.
Screens, Terminal and Window Processing	
%F=	Frame characters for window.
%K	Simulate PF- and PA-keys.
%K*nn*, **%KP***n*	Simulate PF- and PA-keys.
%KN, **%KO**, **%KS**	Function-key logic for Siemens terminals.
%N	Activate non-conversational mode.
%QS	Simultaneous output of multiple screens.
%RM	Protection of light pen sensitive fields.
%RN	Suppress compression of screen data.
%RO	Activate/deactivate screen optimization.
%T and **%T***ll/cc*	Set cursor position.
%T*	Position cursor outside window.
%T+ and **%T-**	Position cursor on protected field.
%W	Natural window handling.
%Y	Control of PF-key lines.
%*	Disable display of input characters. In batch mode, suppress printing of next input record read.
Colors, Outlining	
%D=	Control of outlining.
%=	Assign colors to fields.
INPUT Statement, Stack	
Key(s)	**Function**
CLEAR	Interrupt current Natural operation; interrupt recording.
CTRL+D	Interrupt current Natural operation.

Case Translation	
%L	Disable lower- to upper-case translation.
%U	Enable lower- to upper-case translation.
Copy/Clear	
%C	Copy current screen into Natural source area.
%CC	Copy data into system variable *COM.
%CS	Copy data to the stack.
%Z	Clear source area.
Language, Messages, Error Processing	
%E=	Activate/deactivate error processing.
%L=	Set language code.
%M	Control of message line.
%MSGSF	Display system error messages in full.
Screens, Terminal and Window Processing	
%F=	Frame characters for window.
%K	Simulate PF- and PA-keys.
%K*nn*, **%KP***n*	Simulate PF- and PA-keys.
%KN, **%KO**, **%KS**	Function-key logic for Siemens terminals.
%N	Activate non-conversational mode.
%QS	Simultaneous output of multiple screens.
%RM	Protection of light pen sensitive fields.
%RN	Suppress compression of screen data.
%RO	Activate/deactivate screen optimization.
%T and **%T***ll/cc*	Set cursor position.
%T*	Position cursor outside window.
%T+ and **%T-**	Position cursor on protected field.
%W	Natural window handling.
%Y	Control of PF-key lines.
%*	Disable display of input characters. In batch mode, suppress printing of next input record read.
Colors, Outlining	
%D=	Control of outlining.
%=	Assign colors to fields.
INPUT Statement, Stack	
RESET+ENTER	Terminate current processing loop.

Terminal Command Key Assignments

The following topics are covered:

- Assigning Terminal Commands to Function Keys
- CLEAR Key - Interrupt Current Operation
- CTRL+D Keys - Interrupt Current Operation
- RESET+ENTER Keys - Attention Interrupt

Assigning Terminal Commands to Function Keys

For enhanced operating convenience, you can assign a frequently used terminal command to a function key. The following methods exist:

- Within a Program, you can assign terminal commands to function keys by using the **statement** SET KEY.
- In the programming environment, you can use the **system command** KEY to assign a terminal commands to a function key.
- Function-key assignments can also be made by the Natural administrator via the **profile parameter** KEY.

Note:
Assignments made with the system command KEY are totally independent of assignments made with a SET KEY statement in a program.

CLEAR Key - Interrupt Current Operation

Pressing the CLEAR key has the same effect as the terminal command "%%".

You can also use the CLEAR key to interrupt a recording that is being played back in step mode. Further information on recording is provided in the section Recording in the Natural Utilities documentation.

CTRL+D Keys - Interrupt Current Operation

Note:
This function is only available on mainframe terminals.

Pressing the keys CTRL+D has the same effect as the terminal command "%%".

RESET+ENTER Keys - Attention Interrupt

Note:
This function is only available under Com-plete on terminals connected via a local controller.

Pressing RESET and then ENTER terminates the current processing loop. The loop must contain a database access statement.

% - Continuation Indicator for INPUT in Batch

% - Continuation Indicator for INPUT in Batch

Note:
This command is for Mainframes only.

Command Syntax

%

When a "%" is used as the last non-blank character of a data record in batch mode (applies only on mainframe computers), it causes the next record to be treated as a continuation record.

See the INPUT statement (in the Natural Statements documentation) for further information.

%% and %. - Interrupt Current Operation

%% and %. - Interrupt Current Operation

Note:
The terminal commands "%%" and "%." will be ignored if the profile parameter ESCAPE is set to OFF.

Command Syntax

%%

These terminal commands can be used to interrupt the current operation.

The following topics are covered below:

- %% in Online Mode
- %% in Batch Mode
- %. in Online and Batch Mode

%% in Online Mode

If you enter "%%" in any field on the screen, the currently active Natural program will be terminated immediately and Natural will return to command input mode.
If you enter "%%" in command input mode, the Natural session will terminated (equivalent to the system command FIN).

"%%" has the following effects:

- The contents of the Natural stack will be deleted.
- Any logical database transaction currently being processed is backed out.
- The source program currently in the work area of the editor will not be affected.

%% in Batch Mode

Note:
This command is for Mainframes only.

In batch mode, "%%" may be used to set restart points in the input files and thus ensure the synchronisation of the input files in the case of an error.

Influence of Profile Parameter CC

Command	Function
CC=ON	If the profile parameter CC (see the Natural Parameter Reference documentation) is set and an error occurs during the compilation/execution of a Natural program in batch mode, the input data stream for the SYNIN and OBJIN input files will be flushed until a line containing "%%" in the first two positions is encountered (if no "%%" is encountered, it will be flushed until the end-of-file is reached). In addition, the contents of the Natural stack will be deleted. If more data are available in the input stream, Natural resumes processing with the line after "%%".
CC=OFF	Any "%%" in the input data will be ignored.

%. in Online and Batch Mode

%.

Online, "%." is the same as "%%", except that the Natural stack is not deleted.

On mainframe computer in batch mode, "%." causes reading of input values for the current INPUT statement to be terminated.

%* - Inhibit Character Display

Command Syntax

%*

The following topics are covered below:

- %* in Online Mode
- %* in Batch Mode

%* in Online Mode

This command may be used when entering sensitive data (e.g., passwords). %* causes all fields on the current screen to be non-displayable.

When used with the SET CONTROL statement, %* causes all fields on the next screen to be non-displayable.

%* in Batch Mode

Note:
This command is for Mainframes only.

In batch mode, %* causes printing of the next input data record to be suppressed. The input line after the line which contains %* will not be printed. This is useful, for example, for suppressing the printing of passwords.

Example of %* in Batch Logon to Natural Security:

```
//CMSYNIN DD *
%*
SYSSEC,DBA,DBA
...
```

The printing of all input data in batch may be controlled with the ECHO profile parameter (see the Natural Parameter Reference documentation).

%.P - Delete First Entry from Stack

Note:
This command is for Mainframes only.

Command Syntax

```
%.P
```

This command deletes the topmost entry from the Natural stack.

For further information on the stack, see the Natural Programming Guide.

%.S - Read Stack Entry without Deleting it

Note:
This command is for Mainframes only.

Command Syntax

%.S

Normally, commands/data are deleted from the stack as soon as they have been read from the stack.

This command causes the next INPUT statement to read the topmost entry from the stack without deleting it. The entry is treated as input data, no matter whether it actually is data or a command.

This allows you check a stack entry and then, depending on its content, decide whether to process it or not.

To check the contents of the stack, you can also use the system variable *DATA.

For further information on the stack, see the Natural Programming Guide.

%/ - End-of-File

Note:
This command is for Mainframes in batch mode only.

Command Syntax

```
%/
```

When entered in the first two positions of an input record being read with an INPUT statement in batch mode, this command causes an end-of-file condition.

%+ and %- - Natural Connection

Note:

- These commands are not available with Natural for Windows.
- They only apply if Natural Connection is installed.

Command Syntax

These terminal commands are used to activate/deactivate Natural Connection.

Command	Function
%+	This command causes the Natural system variable *DEVICE to be set to "PC". This permits usage of the Natural Connection system.
%+N	Same as %+. In addition, this command causes no field names to be sent when uploading/downloading data. **Note**: This command is for Mainframes only.
%-	This command causes the Natural system variable *DEVICE to be set to the setting in effect prior to execution of a %+ command.

See the Natural Connection documentation for further information.

%<TECH - Display Technical Information

Command Syntax

```
%<TECH
```

This terminal command corresponds to the system command TECH.

%<TEST - Invoke the Natural Debugger

Note:
This command is for Mainframes only.

Command Syntax

```
%<TEST
```

With this terminal command, you invoke the Natural Debugger which is one of the Natural Test Utilities.

See also system command TEST in the Natural System Command Reference documentation.

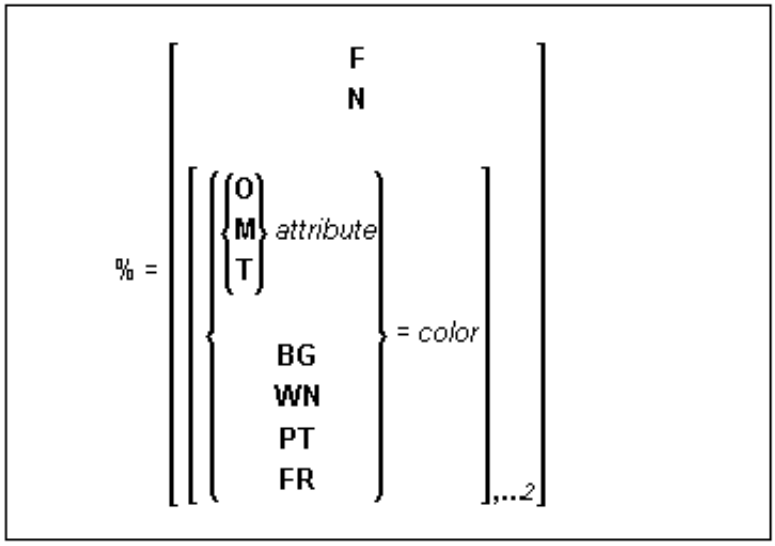
For details on the Natural Debugger see the relevant documentation.

%= - Assigning Colors to Fields

%= - Assigning Colors to Fields

Screen Design (Assigning Colors to Fields - Terminal Command %=) in the Natural Programming Guide.

Command Syntax



With this command, you can assign colors to field attributes for programs that were originally not written for color support. The command causes all fields/text defined with the specified attributes to be displayed in the specified color.

If predefined color assignments are not suitable for your terminal type, you can use this command to override the original assignments with new ones.

You can also use this command within Natural editors, for example to define color assignments dynamically during map creation.

Command	Function
General settings:	
blank	Clear color translate table.
F	Newly defined colors are to override colors assigned by program.
N	Color attributes assigned by program are not to be modified.
Field types:	
O	Output fields (AD=O).
M	Input fields; that is, input-only fields (AD=A) and modifiable fields (AD=M).
T	Text constants.
Possible field attributes:	
B	Blinking
C	Italic
D	Default
I	Intensified
U	Underlined
V	Reverse video
A color can also be assigned to the following parts of a screen:	
BG	Background
WN	Foreground (that is, fields for which no color is defined)
PT	Default page title
FR	Frame of a window
Possible colors:	
BL	Blue
GR	Green
NE	Neutral
PI	Pink
RE	Red
TU	Turquoise
YE	Yellow

Example:

%=TI=RE,OB=YE

This example assigns color red to intensified text fields and color yellow to all blinking output fields.

%A - Play Back a Recording

Note:
This command is for Mainframes only.

Command Syntax

%A *name*

With the terminal command **%A***name*, you cause the recording saved under that *name* to be executed again. This requires that your current library is the one in which the recording is stored.

If you issue the command **%A***name* while a session is being recorded, the recording specified with **%A***name* will not be executed but the command **%A***name* is included into the source that is being recorded. Thus, you can execute a recording from within another recording and concatenate a series of recordings to one another. However, you cannot have nested recordings; the execution of the recording that contains the **%A***name* command stops after that command and is not resumed when the execution of name finishes.

Further information on recording is provided in the section Recording in the Natural Utilities documentation.

%B - Activate/Deactivate a Recording

Note:
This command is for Mainframes only.

Command Syntax

%B *name*

The recording process is activated with the terminal command **%B***name*, and deactivated with the terminal command **%B**.

Command	Function
%B*name*	Activates the recording process. All subsequent actions will be recorded. The name you specify is the name under which the recorded data will be saved. Names of recordings must be unique within a library.
%B	Deactivates the current recording process. The recorded source is automatically saved, and you can then play it back as often as you wish. You may also use **%B** to insert additional actions into the recording: after you have interrupted the playback of a recording with the CLEAR key, enter the command **%B**, and all action you perform will be inserted into the source of the recording until you enter **%B** again. Then the execution of the recording will be resumed.

Further information on recording is provided in the section Recording in the Natural Utilities documentation.

%B= - Specify Library for a Recording

Note:
This command is for Mainframes only.

Command Syntax

%B=*library-name*

With the terminal command **%B=***library-name*, you specify the library in which all subsequent recording actions are to be stored.

If you activate the recording process without having specified a library, the name of the library in which the recording is stored will be the same as the value of the system variable *INIT-USER at the time when the recording process is activated.

When you log on to another library during a session being recorded, the library in which the recording is being stored will remain the same (that is, either the one specified with **%B=** or the *INIT-USER library); this means that one recording may record actions across multiple applications.

Further information on recording is provided in the section Recording in the Natural Utilities documentation.

%C - Copying Contents of Page Buffer

Command Syntax

%C

This terminal command is used to copy the contents of the page buffer to the next available lines in the Natural source work area.

The page currently displayed by Natural will be copied into the Natural source work area. The page content will be written to the next free location in the source work area, where it can be modified using the Natural editor.

To clear the source work area before copying the page, the %Z terminal command may be used.

Notes:

- %C should not be used in an editor session. Modifications made to the source area outside of editor content are not recognized by the editor.
- The page buffer (the logical output from Natural) is not necessarily the same as the screen buffer which is displayed on the screen.

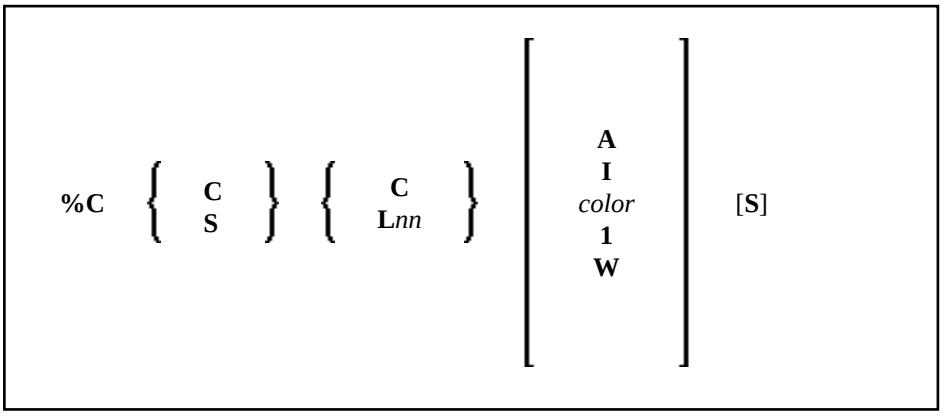
Example:

```
FOR I = 1 TO 10
  WRITE I
  SET CONTROL 'C'
END-FOR
END
```


%CS and %CC - Copying Data to Stack or *COM

See also Dialog Design, Copying Data from a Screen, in the Natural Programming Guide.

Command Syntax



* **1** and **W** cannot be specified with **L***nn*.

With this terminal command, you can copy parts of a screen into the Natural stack (%CS) or into the system variable *COM (%CC). The protected data from a specific screen line are copied field by field (except with option "A"; see below).

The second letter in the command determines where the data are copied to:

- **%CC...** copies the data into the system variable *COM.
- **%CS...** copies the data into the Natural stack. The data are placed on top of the stack as input data (as with a STACK TOP DATA statement).

The third letter in the command determines the line from which the data are copied:

- **%CCC** and **%CSC** copy all protected data from the line in which the cursor is positioned, beginning from the field in which the cursor is positioned.
- **%CCL***nn* and **%CSL***nn* copy all protected data from line number nn.

Moreover, you have the following options:

- **%C...A** copies all of a line, that is, not only the protected data, but also the modifiable fields; the line is not copied field by field, but as a whole (including field attributes).
- **%C...I** copies only the intensified fields from a line.
- **%C...***color* copies only the fields of that color from a line.
- **%C...C1** copies only one field, namely the field in which the cursor is positioned (regardless of its attributes). (**%C...L***nn***1** is not possible.)
- **%C...CW** copies only the word (as delimited by blanks or special characters within a field) over which the cursor is positioned. (**%C...L***nn***W** is not possible.)
- **%C...S** causes Natural to "stay" on the screen from which the data are copied, when the command is executed. This allows you to copy several different data from a screen, before you process the data.

When you enter the command directly as %C... (or assign it to a PF-key), it applies to the *physical screen* within the active window. Exception: On mainframe computers, it applies to the entire *physical screen*; this means that you can copy from the entire screen, no matter whether the data you wish to copy are inside or outside an active window.

When you issue the command via a SET CONTROL statement, it applies to the *logical page* built by Natural. In conjunction with the stay option, this allows you to copy all data you need from an entire logical page (which may be larger than the actual physical screen), before you process these data.

%D - Activate Keyword/Delimiter Mode

Note:
This command is for Mainframes only.

Command Syntax

%D

This command is used to activate keyword/delimiter mode. This mode is recommended for processing in batch mode.

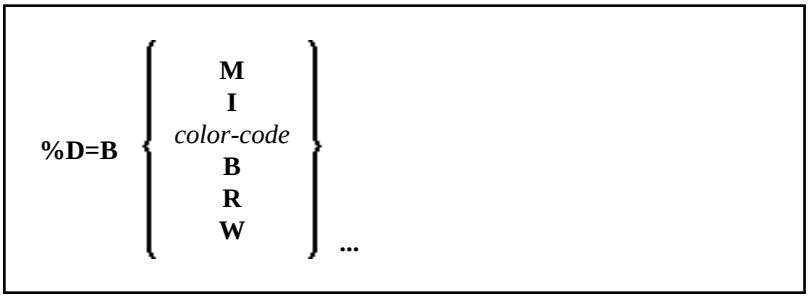
See the INPUT statement in the Natural Statements documentation for additional information.

%D= - Control of Outlining

Notes:

- This command is for Mainframes only.
- The outlining feature is only available on certain types of terminals, usually those which also support the display of double-byte character sets.

Command Syntax



The terminal command %D=B is used to control outlining.

Outlining (boxing) is the capability to generate a line around certain fields when they are displayed on the terminal screen. Drawing such "boxes" around fields is another method of showing the user the lengths of fields and their positions on the screen.

The %D=B command provides the following options:

Command	Function
%D=BM	This command causes all physically modifiable fields (AD=M or AD=A) to be displayed with outlining. If this command is repeated, this feature will be disabled again: in other words, it will return to the default state.
%D=BI	This command causes all intensified fields (AD=I) to be displayed with outlining.
%D=B*color-code*	This command causes all fields of the specified color to be displayed with outlining. For valid color codes, see the session parameter CD.
%D=BB	This command causes "big boxes" to be generated: if several fields displayed underneath one another have identical display characteristics, one big box is drawn around all of them. Normally (if %D=BB is not specified) an individual box is drawn around each individual field.
%D=BR	This command causes all boxes to be reset; that is, no field will be displayed with outlining.
%D=BW	This command causes all windows which are to be displayed with frame (see the DEFINE WINDOW statement in the Natural Statements documentation) to be displayed with outlining instead of frame.

When outlining is active, filler characters in modifiable fields are not displayed, because they serve the same purpose as outlining, namely showing the user the position and length of a field.

%DUE - Dump for Specific Error

Command Syntax

The terminal command %DUE is used within a session to specify Natural error numbers for which a storage dump shall be taken. This may be helpful to get a dump for the analysis of a specific error situation by Software AG personnel. See also profile parameter DUE.

The %DUE command provides the following options:

Command	Function
%DUE+*nnnn*	Adds an error number for which a storage dump is to be taken. If the number is greater than 9999, a NAT1134 error message is issued.
%DUE-*nnnn*	Removes a previously specified error number from the table. If the number is greater than 9999, a NAT1134 error message is issued.
%DUE-ALL	Removes all previously specified error numbers from the table.

If an error occurs which has been specified by %DUE, a program check is forced. If profile parameter DU=OFF is set, it will be changed to DU=ON. For further processing, the DU profile parameter setting is honored.

Examples:

```
%DUE+82
%DUE+80
%DUE-82
%DUE-ALL
```


%E - Display Screen Captures

Note:
This command is not available with Natural for Windows.

Command Syntax

%E

With this command, you can display any screens that have been captured with the NATPAGE screen paging function (NATPAGE utility).

This command suspends normal processing and enters the display mode of the NATPAGE function: A list of all screens captured with NATPAGE (by using the terminal commands %I and %P) is displayed; including screen number, time of capturing, and map name (if the screen is a map). From the list you can then select the screens you wish to have displayed. You cannot enter any data on the displayed screens.

When a screen is captured with %I or %P, two items of information are superimposed on the screen: the time when the screen was captured, and a screen number (the screens are numbered sequentially in the order in which they are captured).

Natural for Mainframes:
In NATPAGE display mode, an input field (CMD) is provided in which you can enter the following scrolling commands:

Command	Function
TOP or **T**	Displays the first screen which was captured.
BOT or **B**	Displays the last screen which was captured.
nnn	Displays the screen with screen number nnn.
+nnn	Scrolls nnn screens forward.
-nnn	Scrolls nnn screens backward.
.	Terminates display mode.

If you attempt to scroll to a screen number which does not exist, either the first or the last screen is displayed, depending on the scrolling direction.

Natural for UNIX:
A window will be displayed in which you can scroll through the list of screens captured. With ENTER, you can select a screen for display. With ESC, you leave the function.

See also the terminal commands %P, %O, %S, and %I.

%E= - Activate/Deactivate Error Processing

Command Syntax

```
%E=  { ON  }
     { OFF }
```

With the terminal command %E=OFF any error transaction and ON ERROR processing is switched off, with %E=ON error transaction and ON ERROR processing is switched on again.

Command	Function
%E=OFF	Switches any error transaction (as identified by the system variable *ERROR-TA or defined in Natural Security) and ON ERROR processing off; any error that occurs will then be handled by normal Natural error processing. This may be used to locate an error in your applications' error processing if the structure of the application with various error-handling procedures on different levels makes it impossible for you to find out where exactly an error originally occurred.
%E=ON	Switches error transaction and ON ERROR processing on again.

%F - Activate Forms/Screen Mode

Note:
This command is for Mainframes only.

Command Syntax

%F

This terminal command activates the forms/screen mode.

See the INPUT statement (in the Natural Statements documentation) for additional information on forms/screen mode.

%F= - Frame Characters for Window

Note:
On graphical user interfaces, this command is ignored.

See also:

Screen Design (Windows) in Natural Programming Guide.

Command Syntax

%F=*chv*

With this terminal command, you can define the characters to be used in the frame of a window.

c	The first character will be used for the four *corners* of the window frame.
h	The second character will be used for the *horizontal* frame lines (top and bottom frame lines).
v	The third character will be used for the *vertical* frame lines (left and right frame lines).

Example: The command `%F=+-!` would cause the window frame to look like this:

```
+------------------------+
!                        !
!                        !
!                        !
!                        !
+------------------------+
```

For further information on window processing, see the DEFINE WINDOW statement (in the Natural Statements documentation).

%FM - Numeric Edit Mask Free Mode

Command Syntax

This command is used to activate/deactivate edit mask free mode, a special capability to allow literals to be omitted during input into a field with a numeric edit mask.

Command	Function
%FM-	With this command, you switch edit mask free mode off.
%FM+	With this command, you switch edit mask free mode on.

The default setting at session startup is provided with Profile Parameter EMFM.

See the INPUT statement (in the Natural Statements documentation) for additional information on the edit mask free mode.

%G - Set Playback Mode for a Recording

Note:
This command is for Mainframes only.

Command Syntax

```
%G { ON  }
   { OFF }
```

A recording can be played back in two modes:

- **Background Mode**
  In background mode, the entire recording is played back invisibly; that is, all the actions of the recording are carried out without anything being displayed to you on the terminal screen during the execution of the recording. Neither can you interrupt a recording that is played back in background mode (unless the recording contains the terminal command %R).
- **Step Mode**
  In step mode, a recording is played back step by step, and all actions are displayed to you. By pressing ENTER you proceed from one step to the next. In step mode, it is also possible for you to interrupt the recording by pressing the CLEAR key.

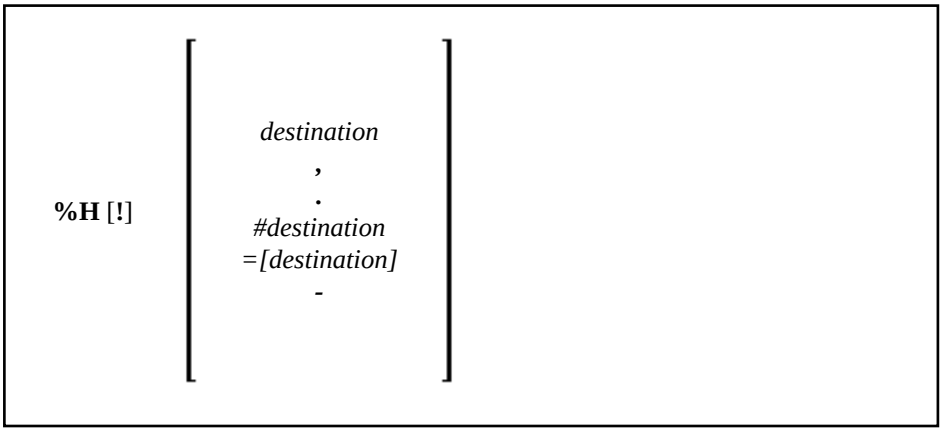
By default, a recording is played back in background mode.

Command	Function
%GON	With this command, you switch step mode on.
%GOFF	With this command, you switch back to background mode.
%G	With this command, you can toggle between the two modes.

Further information on recording is provided in the section Recording in the Natural Utilities documentation.

%H - Hardcopy Output

Command Syntax



This command is used to produce hardcopy output (valid only if the hardcopy feature is implemented).

Natural under CMS:
Specifying %HL produces a file called NATURAL LISTING A.

The **%H** command produces output from Natural reports and communication screen layouts on a hardcopy printer. By default, a %H command is effective for the current logical output (that is, the current window without message line, function-key lines and statistics line/infoline). However, you can also print the current screen content (see **%H,** and **%H.** below).

You have the following options:

Command	Function
%H	A window is displayed listing all available printers; from this list you select the printer on which the hardcopy is to be printed. **Natural for Mainframes:** No selection window is displayed; instead, the printer specified with the profile parameter HCDEST is used. **Natural for Windows:** Printer "LPT1" will be used by default if you do not specify one.
%H*destination*	The hardcopy will be output on the specified *destination*. The *destination* can be 1 to 8 characters long.
%H!	**Note:** This command option is only valid under UNIX and Windows. %H! applies to the page currently displayed on the screen.
%H!*destination*	**Note:** This command option is only valid under UNIX and Windows. SET CONTROL 'H!*destination*' is valid only for the current page (that is, the page that was output before the SET CONTROL statement was executed).
%H,	All subsequent %H commands apply to the current *screen* (only possible on mainframe computers).
%H.	All subsequent %H commands apply to the current **logical page** (applies by default).
%H=	Every time ENTER is subsequently pressed, a hardcopy is produced (toggle switch).
%H=*destination*	With **%H=***destination*, you specify the hardcopy device to which **all** logical output is to be output. This feature may be used, for example, to log a sequence of output for administrative, debugging or educational purposes.
%H#*destination*	With **%H#***destination*, you can have the hardcopy output routed to a special destination, as defined with a DEFINE PRINTER statement, for example, to the Natural source area, to Con-nect, or to the INFOLINE.
%H-	Stops hardcopy printing immediately. **Note:** When a SET CONTROL 'H-' statement is executed, data which were already written to the page buffer but have not been output yet, will not be routed to the printer. To also print these data, you have to code an EJECT statement before the SET CONTROL 'H-' statement.

Generally, a %H command (entered anywhere on the screen or used with a SET CONTROL statement) is valid until either the next INPUT statement with modifiable fields is executed, or until the end of the program. However, when %H is entered in an input field, only the current logical page will be printed.

Note:
The EJECT statement does not affect the %H command. The %H command always causes Natural to advance pages.

%I - Capture Current Screen

Note:
This command is not available with Natural for Windows.

Command Syntax

%I

When you enter this command on a screen, the screen will be captured by the NATPAGE screen paging function (NATPAGE utility).

The maximum number of screens that can be captured is determined by the session parameter PD. If this number is exceeded, each additional screen will overwrite one that has already been captured, beginning with the one captured first.

To display screens that have been captured with the screen paging function, use the terminal command %E.

See also the terminal commands %P, %O, %S, and %E.

%J - Invoke Helproutine

Command Syntax

%J *helproutine*

This terminal command can be used to invoke an interactive helproutine.

When %J is used after interrupting the playback of a recording (applies only on mainframe computers), the playback will be resumed after the helproutine has been executed (refer to the section Recording in the Natural Utilities documentation).

If %J is used when a function invoked by a system command is active, Natural will search for the specified helproutine in the active
library of the system command or in a library that has been defined as a steplib for the system command.

%KN/%KO/%KS - Siemens Function-Key Logic

Notes:

- These command are for Mainframes only.
- They are only applicable to Siemens terminals.

Command Syntax

The following commands are available:

%KN	With terminal types 8160, 974n, 9750 - 9755	This command causes the literals "%K1" to "%K20" to be loaded to the keys P1 to P20.
	With terminal types 9756, 9758, 976n	This command causes the send-key codes "F1" to "F20" to be loaded to the keys P1 to P20.
%KO	This command causes the literals "01" to "20" as well as the send-key code "F5" to be loaded to the keys P1 to P20.	
%KS	This command causes the literals "A" to "T" as well as the send-key code "F5" to be loaded to the keys P1 to P20.	
%KNN **%KON** **%KSN**	If an "N" is specified after the terminal commands %KN, %KO or %KS, only the corresponding function-key mode (KN, KO or KS mode) will be activated, but no values will be loaded to the P-keys.	

See also Natural under BS2000/OSD (in the Natural Operations for Mainframes documentation).

%K and %KP - Simulate PF- and PA-Key

%K and %KP - Simulate PF- and PA-Key

Command Syntax

These terminal commands can be used to simulate the terminal function (PF, ENTER) and program attention (PA) keys.

Command	Function
%K*nn*	Simulates the terminal function key numbered *nn* (PF1 to PF24). This permits PF-keys 13-24 to be assigned to PF-keys 1-12, or the activation of PF-keys not available on the keyboard used. **Natural for Mainframes:** This terminal command also makes function keys available in batch mode.
%K0	Simulates the ENTER key.
%KP*n*	Simulates the program attention key numbered *n* (PA1 to PA3) (see %K*nn*).

%L - Disable Lower- to Upper-Case Translation

Command Syntax

%L

This command prevents that lower-case characters are translated to upper-case by Natural.

%L influences the interactive input, which is entered e.g. with a Natural INPUT statement. It does not, however, influence the input from the stack.

Natural for Mainframes:
You should also ensure that the TP monitor used does not perform any translation before data are passed to Natural.

See also the terminal command %U.

%L= - Set Language Code

%L= - Set Language Code

Command Syntax

%L=*nn*

With the terminal command **%L=***nn*, you can set the language code nn to be used by Natural.

For a list of possible language codes, see the system variable *LANGUAGE.

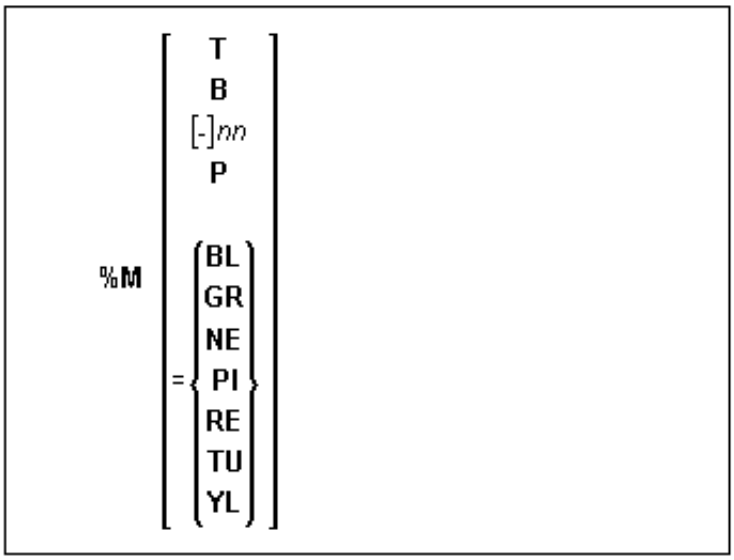
%M - Control of Message Line

Note:

- This command is ignored in batch mode.
- This terminal command also applies to Natural's NEXT (or MORE) line.

Control of the Message Line - Terminal Command %M (see Screen Design in Natural Programming Guide)

Command Syntax



With this terminal command, you can control the position, the protection mode and the color of the Natural message line.

The following topics are covered below:

- Message Line Positioning
- Message Line Protection
- Message Line Color

Message Line Positioning

Command	Function
%MB	causes the message line to be the bottom line of the screen.
%MT	causes the message line to be the top line of the screen.
%M	causes the current message line position to be switched from the top line to the bottom line of the screen (or vice versa), or from line *nn* to the bottom line.
%M*nn*	causes the message line to be placed on line *nn* of the screen.
%M-*nn*	causes the message line to be placed on the *nn*th line from the bottom of the screen. If the line number *nn* or *-nn* is not within the current screen, the message line will not be displayed.

Message Line Protection

Command	Function
%MP	switches from protected to unprotected message line or vice versa.

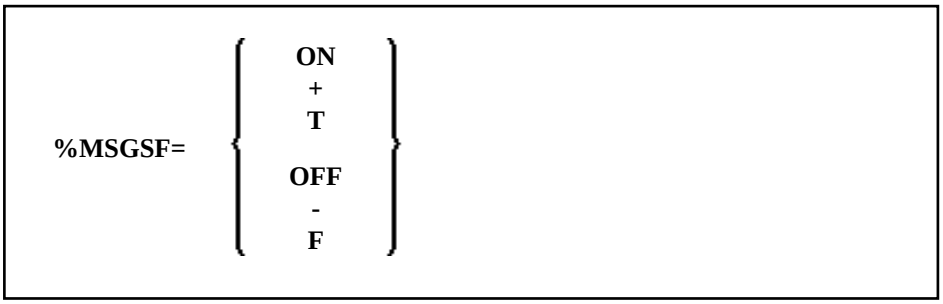
Message Line Color

Command	Function
%M=*color-code*	When a color screen is used, this terminal command causes the message line to be displayed in the specified color (for an explanation of color codes, see the session parameter CD).

%MSGSF - Display System Error Messages in Full

Note:
This command is for Mainframes only.

Command Syntax



By default, a Natural system error message consists of: the name of the program and the number of the line that caused the error, followed by the actual text of the message. Depending on the size of the window in which the message is displayed, the actual text may be truncated. With the terminal command %MSGSF, you can avoid such truncation.

The following command options are available:

Command	Function
%MSGSF=ON **%MSGSF=+** **%MSGSF=T**	System error messages will be displayed in full; that is, program name, line number and actual message text will be displayed (this is the default).
%MSGSF=OFF **%MSGSF=-** **%MSGSF=F**	System error messages will be displayed in short form; that is, only the actual message text will be displayed (but not the program name and line number).

Notes:

- Instead of "ON", you can also specify "+" or "T" (true).
  Instead of "OFF", you can also specify "-" or "F" (false).
- The display format of system error messages can also be controlled with the profile parameter MSGSF (see the Natural Parameter Reference documentation).

%N - Activate Non-Conversational Mode

Command Syntax

```
%N
```

This terminal command is used with a SET CONTROL statement and causes the next screen to be displayed without requiring any user response for processing to continue; that is, after the screen has been displayed, processing will continue immediately without waiting for any user input.

This command may be used to send messages about the progress of program execution to the user. The command only applies to the next output screen (that is, the screen that is output when the next I/O is executed).

Under IMS/TM:
This terminal command does not apply if an IMS scratch-pad area is used.

%O - Deactivate Screen Capturing

Note:
This command is not available with Natural for Windows.

Command Syntax

%O

With this command, you deactivate the capturing of screens by the NATPAGE screen paging function (NATPAGE utility) as activated by the terminal command %P.

The current screen is captured. All screens captured since the last %P command are retained.

See also the terminal commands %E, %I, %P and %S.

%P - Capture Screen Sequence

Note:
This command is not available with Natural for Windows.

Command Syntax

%P

With this command, you activate the NATPAGE screen paging function (NATPAGE utility) to capture the current screen and all subsequent screens.

The maximum number of screens that can be captured is determined by the session parameter PD. If this number is exceeded, each additional screen will overwrite one that has already been captured, beginning with the one captured first.

All screens that have been captured with previous %P and %I commands will be deleted when you issue a %P command.

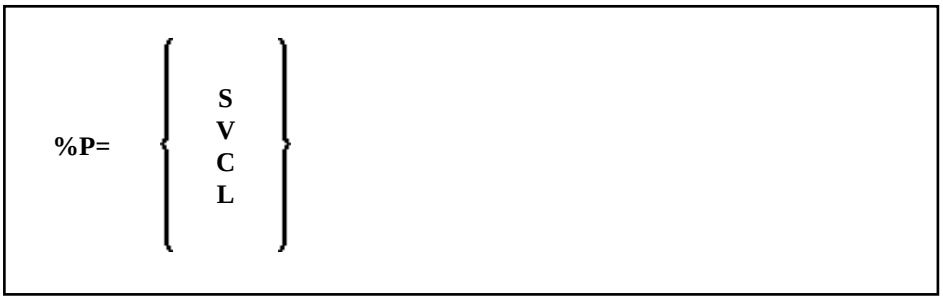
To display screens that have been captured, you use the terminal command %E.

See also the terminal commands %E, %I, %O and %S.

%P= - CALL Options

Note:
The following commands are for Mainframes only.

Command Syntax



The commands %P=S, %P=V and %P=C may be used to set special options which apply when a Natural program calls a non-Natural program (via a CALL statement) under CICS. In all other environments, these commands are ignored.

For details on calling non-Natural programs, see the CALL statement in the Natural Statements documentation.

Each %P= command applies only to the next call.

The following command options are covered below:

- %P=S - Standard Linkage for Call
- %P=V - Roll-Out for Call
- %P=C - Pass Parameter Values instead of Address
- %P=L - Call of LE Dynamic Main Program

%P=S - Standard Linkage for Call

Note:
This command only applies under CICS.

Normally, when a Natural program calls a non-Natural program under CICS, the call is accomplished by an "EXEC CICS LINK" request.

If standard linkage is to be used for the call instead, issue the terminal command %P=S. In this case, the called program must adhere to standard linkage conventions with standard register usage.

%P=V - Roll-Out for Call

Note:
This command only applies under CICS.

Normally, when a Natural programs calls a non-Natural program and the called program issues a conversational terminal I/O, the Natural thread is blocked until the user has entered data.

Under CICS, %P=V can be used to prevent the Natural thread from being blocked: if this terminal command is specified, the parameter data passed from the Natural program to the called program are copied out of the Natural thread, and the thread is rolled out before the call. The thread is then available for another user. Upon return from the called program to the calling Natural program, the thread is rolled back in again, the (modified) data area is copied into the thread, and Natural processing continues.

Note:
Only the parameters specified in the CALL statement are copied out of and back into the thread.

%P=C - Pass Parameter Values instead of Address

Note:
This command only applies under CICS.

Normally, when a Natural program calls a non-Natural program under CICS, the address of the CALL parameter address list is passed in the COMMAREA. If you wish the parameter values themselves, rather than the address of their address list, to be passed in the COMMAREA, issue the terminal command %P=C before the call.

This makes it possible, for example, to use DPL for called CICS programs: A CICS program that resides in another CICS region can only be called with %P=C; as addresses within the "calling" region cannot be accessed by the "called" region, the parameter values have to be passed instead.

When %P=C is used, no parameters are passed in the TWA, but only parameter values in the CICS COMMAREA. All parameters of the CALL parameter list are copied adjacent to one another, regardless of their alignment. The resulting COMMAREA length is the sum of the individual parameter lengths (this has to be taken into consideration when determining the number of array occurrences to be passed). On return from the called program, the parameters are copied back.

When overlapping fields are passed or the same field is passed more than once, these fields should be made "read-only" for the called program; otherwise, unpredictable results may occur when parameter values are returned to the calling program.

For %P=C, the restriction applies that group arrays cannot be passed:

```
01 #GROUP (2)    02 #FIELD1 (A1)    02 #FIELD2 (P7)
```

Either pass them as individual arrays:

```
01 #GROUP    02 #FIELD1 (A1/2)    02 #FIELD2 (P7/2)
```

Or redefine them:

```
01 #GROUP 01 REDEFINE #GROUP    02 #ARRAY (A1/10)
```

and specify the array name in the CALL statement.

Note:
If both %P=S and %P=C are issued for a call, %P=C will be ignored.

%P=L - Call of LE Dynamic Main Program

Note:
This command only applies if the option for the support of the IBM Language Environment (LE) calling conventions has been set at the installation of Natural.

This command causes control to be returned to Natural after a call to an LE dynamic main program.

By default, when an LE dynamic main program is invoked, control is not returned to Natural after the invoked program has been processed. For control to be returned after that program has been processed, you have to use %P=L before invoking the program.

For information on how Natural supports IBM Language Environment (LE) subprograms, see LE Subprograms in the Natural Operations for Mainframes documentation.

%Q - Suppress Map Printing in Batch Mode

Command Syntax

%Q

In interactive processing, the %Q command is ignored. When issued in batch mode (applies only on mainframe computers), the %Q command causes the output of maps, or screens produced by INPUT statements, to be suppressed.

SET CONTROL ’Q’ (online or batch) causes the next INPUT statement **not** to be processed. This may be used, for example, if at the end of a helproutine the processing is to continue without the user having to press ENTER upon return from the help to the map.

%QO - Suppress Pseudo-Conversational Output

Note:

- This command is for Mainframes only.
- It is only applicable under CICS.

Command Syntax

%QO

In a Natural session running under CICS in pseudo-conversational mode, you can switch to another CICS transaction by issuing the statement CALL ’CMTASK’. To enable CICS to start the other transaction, a terminal I/O, i.e. an INPUT statement, is required after the CALL statement.

To suppress the screen output of that INPUT statement (which would be overwritten immediately by the started transaction anyway), you issue a SET CONTROL ’QO’ statement before the INPUT statement.

%QS - Simultaneous Output of Multiple Screens

Command Syntax

%QS

With this command, you can display multiple screens simultaneously.

%QS causes the next screen I/O not to be executed. The corresponding output screen is kept internally until the following I/O, when it is displayed together with the next screen. Therefore %QS only makes sense if the second output screen is a window, that is, if it does not entirely overlay the first screen that was suppressed with %QS.

Example:
You can suppress the output of a screen A with %QS; the next screen B is a window which partially overlays screen A (perhaps a help window for one of the fields on screen A); with the next screen I/O, the window B and the "underlying" screen A are displayed simultaneously.

A %QS command only applies to the subsequent screen.

Note:
As %QS reduces the number of screen I/Os, it also improves performance.

%R - Repeat INPUT Statement

Command Syntax

%R

This command causes INPUT statement repetition and the output screen to be rebuilt. All output data generated from the beginning of the INPUT statement will be reproduced.

%R in a Recording

Note:
This command is for Mainframes only.

By recording the terminal command %R, you can manipulate a single step in a recording when it is played back.

You may also use %R to overwrite input data in a recording that is being played back.

Further information on recording is provided in the section Recording in the Natural Utilities documentation.

%RM - Protection of Light Pen Sensitive Fields

Note:
This command is for Mainframes only.

Command Syntax

%RM

This command causes all light pen sensitive fields on the screen to be made write-protected, that is, the user can select them with a light pen, but not overwrite their contents. To switch the write-protection off again, issue the command again.

For a field to be light pen sensitive, it must be displayed intensified (session parameter AD=I) or blinking (AD=B), and the first character in the field must be one of the light pen designator characters ("?", ">", "&" or a blank); selecting a field with a light pen causes the designator character to be changed, therefore you can make the processing of fields selected with a light pen dependent on the values of the designator characters.

See also Light Pen Support (in the Natural Operations for Mainframes documentation) and the PEN value of the system variable *PF-KEY.

%RN - Suppress Compression of Screen Data

Note:
This command is not available with Natural for Windows.

Command Syntax

%RN

For the next screen I/O, this command suppresses Natural’s automatic compression of screen data and causes the entire screen to be sent instead.

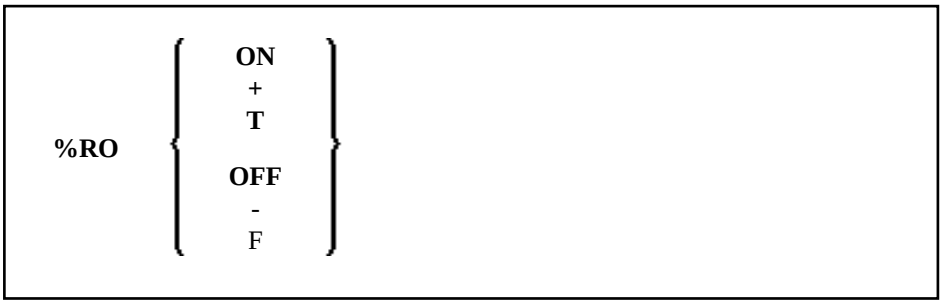
At a screen I/O, Natural usually does not send the entire screen, but only the changed screen data. When a non-Natural program is invoked which causes a screen I/O, Natural in most TP environments recognizes the non-Natural I/O and sends the entire screen upon the next screen I/O. However, in some TP environments, the non-Natural I/O is not recognized by Natural; in such a case, %RN should be used.

%RN only applies to the next screen I/O.

%RO - Activate/Deactivate Screen Optimization

Note:
This command is for Mainframes only.

Command Syntax



With this command you can switch Natural’s automatic screen optimization off and back on again.

Normally, Natural’s screen optimization causes screens to be sent as compressed as possible. If this should conflict with any TP monitor’s screen optimization or hardware limitations, you can use this terminal command to switch Natural’s screen optimization off; screens are then sent in non-compressed form.

If you use the session parameter BX setting BX=L or BX=R, you should switch off Natural’s screen optimization.

The following %RO command options are available:

Command	Function
%ROON **%RO+** **%ROT**	activates screen optimization.
%ROOFF **%RO-** **%ROF**	deactivates screen optimization.
%RO	switches between the two modes.

Note:
Instead of "ON", you can also specify "+" or "T" (true). Instead of "OFF", you can also specify "-" or "F" (false).

%S - Resume Screen Capturing

Note:
This command is not available with Natural for Windows.

Command Syntax

%S

This command is used to resume the capturing of screens by the NATPAGE screen paging function (NATPAGE utility).

The capturing of screens (which is activated with the terminal command %P) can be interrupted with the terminal command %O. After it has been interrupted, you can use the terminal command %S to resume the capturing of screens.

See also the terminal commands %E, %I, %O and %P.

%T - Position Cursor to Top of Screen

Command Syntax

%T

This command positions the cursor at the top lefthand corner of the screen on the next screen output.

Note:
On graphical user interfaces, this command only works if there is an input field (session parameter AD=A or AD=M) at the top lefthand corner.

%Tll/cc - Position Cursor to Line ll, Column cc

Command Syntax

%T*ll/cc*

This command positions the cursor at line *ll*, column *cc* on the next screen output.

The line and column positions are counted beginning with 1 within the current physical screen window. The message line, the function-key lines, and the statistics line/infoline (if active) are not counted as lines in the current window; that is, "%T1/1" always positions to the beginning of the topmost data line in the window.

Notes:

- On some devices, you have to execute the terminal command %T+ before you can place the cursor at any desired position on the screen.
- On graphical user interfaces, %Tll/cc only works if there is an input field (session parameter AD=A or AD=M) at the specified position.

%T+/%T- - Position Cursor on Protected Fields

Notes:

- These commands are only available with Natural for UNIX.
- They are only applicable to Siemens terminals.

Command Syntax

The following command options are available:

Command	Function
%T+	After you issue this command, you can position the cursor anywhere on the screen, including protected fields (however, the protected fields remain write-protected). This may be useful, for example, for action bar processing (see the terminal command %YC).
%T-	With this command, you cancel the effect of the command **%T+**.

%T* - Position Cursor Outside Window

Command Syntax

```
%T*
```

Normally, when a window is active and the window contains no input fields (AD=A or AD=M), the cursor is placed in the top left corner of the window.

This terminal command causes the cursor to be placed in a *COM system variable outside the window when the active window contains no input fields.

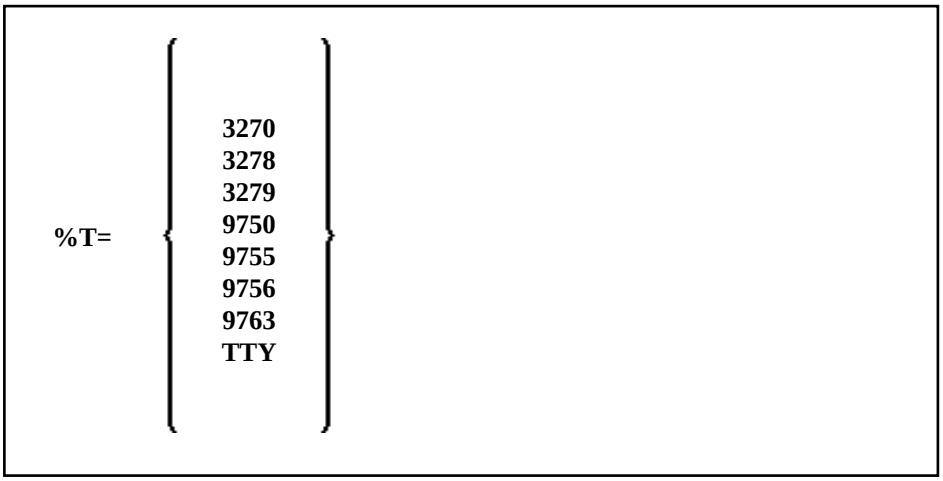
Command	Function
%T*	Switches between cursor placement in system variable *COM outside the window and standard cursor placement within the window.

%T* only applies to the next INPUT statement, and it must be issued **before** the INPUT statement.

%T= - Activate Converter Routine for Device Type

Note:
This command is only relevant if Natural is used on mainframe computers under certain TP monitors.

Command Syntax



This command causes Natural to use the appropriate converter routine for the specified device type.

If the TP environment is not able to inform Natural of the physical terminal type it is working with (in which case the terminal type would be set automatically), this command may be used to specify the terminal type. This allows Natural to build the appropriate attribute sequence to operate the terminal.

The default terminal types used by Natural (if no information is passed from the TP environment) are %T=3270 (for IBM), or the value defined in PDN unless overridden by the parameter T975X (for Siemens; see Installing the Natural TIAM Interface in the Natural for Mainframes Installation Guide).

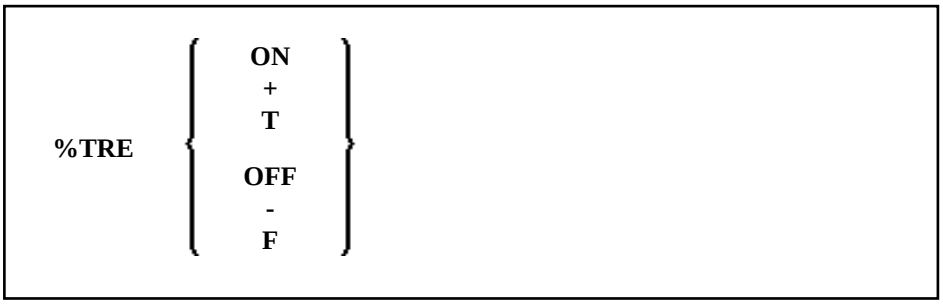
Instead of the %T= command, you can also use the profile parameter Terminal Type TTYPE (see the section Profile Parameters in the Parameter Reference documentation).

Note:
The above diagram lists only the most commonly used terminal types. If your terminal is of another type, ask your Natural administrator to check the NATCONFG module to ascertain whether the terminal type can be specified with %T=.

%TRE - Activate/Deactivate External Trace

Note:
This command is for Mainframes only.

Command Syntax



This command activates/deactivates the external trace function.

Do not use this command without prior consultation of Software AG support. This function is intented primarily for Software AG internal use for debugging purposes. It writes trace data to an external trace dataset depending on the TP environment in which Natural is running.

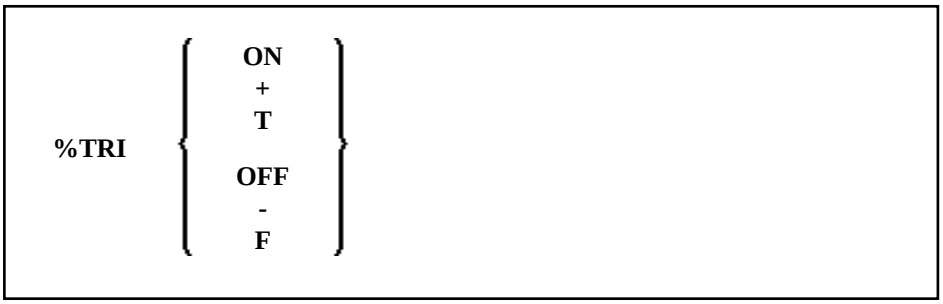
Notes:

- Instead of "ON", you can also specify "+" or "T" (true). Instead of "OFF", you can also specify "-" or "F" (false).
- The external trace function can also be activated/deactivated with the profile parameter ETRACE (see the Natural Parameter Reference documentation).

%TRI - Activate/Deactivate Internal Trace

Note:
This command is for Mainframes only.

Command Syntax



This command activates/deactivates the internal trace function.

This function is intended primarily for Software AG internal use for debugging purposes. It passes trace data to the SYSRDC utility which is described in the Natural Utilities documentation.

Notes:

- Instead of "ON", you can also specify "+" or "T" (true). Instead of "OFF", you can also specify "-" or "F" (false).
- The internal trace function can also be activated/deactivated with the profile parameter ITRACE (see the Natural Parameter Reference documentation).

%U - Translate Lower to Upper Case

Command Syntax

%U

This command causes Natural to translate lower-case characters to upper-case for alphanumeric input data.

Upper-case translation is in effect by default.

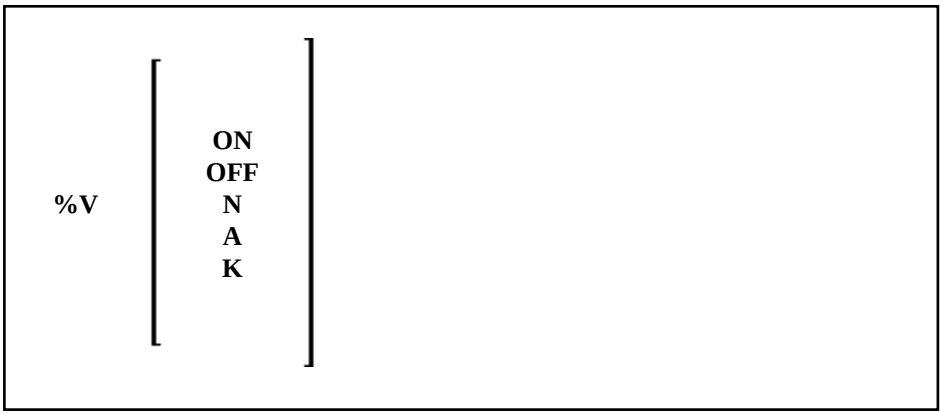
%U influences the interactive input, which is entered e.g. with a Natural INPUT statement. It does not, however, influence the input from the stack.

See also the terminal command %L.

%V - Control of Print Mode

Note:
This terminal command only applies to terminals which support inverse (right-to-left) print mode.

Command Syntax



The options "N", "A" and "K" are only available on mainframe computer terminals.

%VON	activates inverted (right-to-left) print mode.
%VOFF	activates normal (left-to-right) print mode.
%V	switches from normal to inverted print mode, and vice versa.
%VN	switches between the normal print mode and inverted print mode for numeric fields. This may be used for sensitive terminals which do not invert numerics.
%VA	causes alphanumeric fields that are defined with PM=I to be treated as numeric fields. This ensures that numeric values in such fields, which would otherwise be interpreted from right-to-left, will be interpreted correctly from left-to-right. Entering %VA again switches off the effect of %VA again.
%VK	Numeric fields for which a helproutine or edit mask is defined, are internally converted by Natural to alphanumeric format (otherwise a user would not be able, for example, to enter a question mark in such a field to invoke the helproutine). With inverted print mode, however, this internal conversion would cause a numeric value in such a field to be interpreted from right-to-left. For numeric fields that are defined with PM=I and have a helproutine or edit mask, %VK prevents the internal translation to alphanumeric format, thus ensuring the correct interpretation of values in these fields, while an edit mask will still be applied correctly and a question mark can nonetheless be entered in the field. Entering %VK again switches off the effect of %VK again.

For additional information on print modes, see the session parameter PM.

%W - Window Processing

Command Syntax

%W

Note:
You are strongly recommended to use the DEFINE WINDOW statement instead of the %W command.

A Natural window is that segment of a logical page, built by a Natural program, which is displayed on the terminal screen.

The %W command controls the processing of this window.

The command must always be specified with parameters for the various functions as described hereafter. Multiple parameters may be specified with one %W command; they must be specified consecutively without any delimiter characters.

There is always a window present, although you may not be aware of its existence: unless specified differently (with a %W command or DEFINE WINDOW statement), the size of the window is identical to the physical size of your terminal screen.

See also the DEFINE WINDOW statement in the Natural Statements documentation for information on window processing.

There are two types of window commands:

- commands to control the size and position of the window on the physical screen;
- commands to control the position of the window on the logical page created by the program.

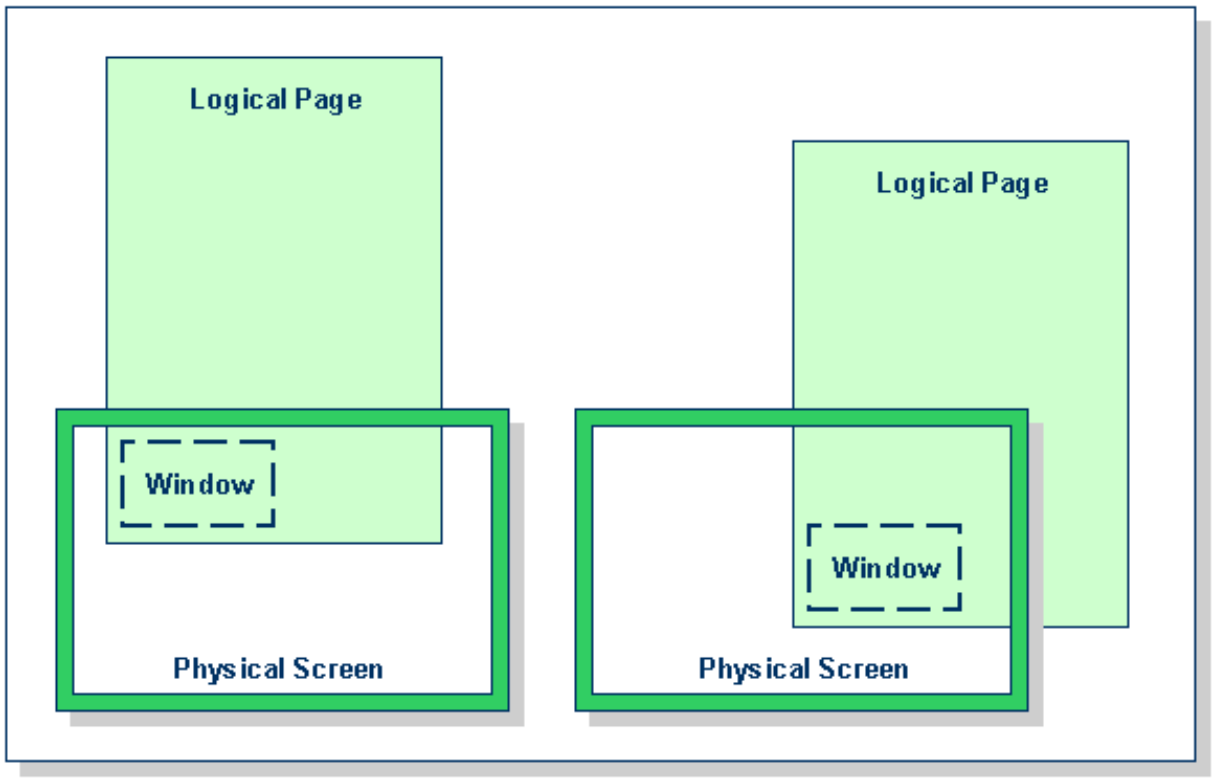
The following topics are covered below:

- Window Size and Position on the Physical Screen
- Window Position on a Logical Page
- %WA and %WZ - Save Screen Image before Window

Window Size and Position on the Physical Screen

The following window commands control the size and position of the window on the physical screen.

When you change the position of the window on the physical screen, the position of the window on the logical page will remain unchanged:



For information on possible window sizes, see the DEFINE WINDOW statement.

Command	Function
%WB	The window size (excluding frame) will be set to physical screen size. If a frame is defined, it will not be visible.
%WB*lll/ccc*	The top left corner of the window will be positioned to line number *lll*/column number *ccc* (lines and columns are counted on the physical screen). The window size will remain unchanged. If the window is too large to be placed at the specified position, it will be placed as close as possible to that position.
%WB0	The window will be positioned to the top left corner of the screen. The window size will remain unchanged.
%W#	The top left corner of the window will be positioned to the cursor position. The window size will remain unchanged. If the window is too large to be placed at the specified position, it will be placed as close as possible to that position.
%W?	The bottom right corner of the window will be set to the cursor position. The top left corner of the window will remain unchanged, and the size of the window will be adjusted accordingly.
%WL*nn*	The line size (horizontal extension) of the window (including frame, if specified) will be set to *nn*. If *nn* is omitted or specified larger than would fit on the screen, the line size will be set to the maximum possible (that is, to the right edge of the screen).
%WC*nn*	The column size (vertical extension) of the window (including frame, if specified) will be set to *nn*. If *nn* is omitted or specified larger than would fit on the screen, the column size will be set to the maximum possible (that is, to the bottom of the screen).

Column size and line size specifications refer to the overall physical size of the window (including frame, if specified), not to the size of what is logically visible inside the window.

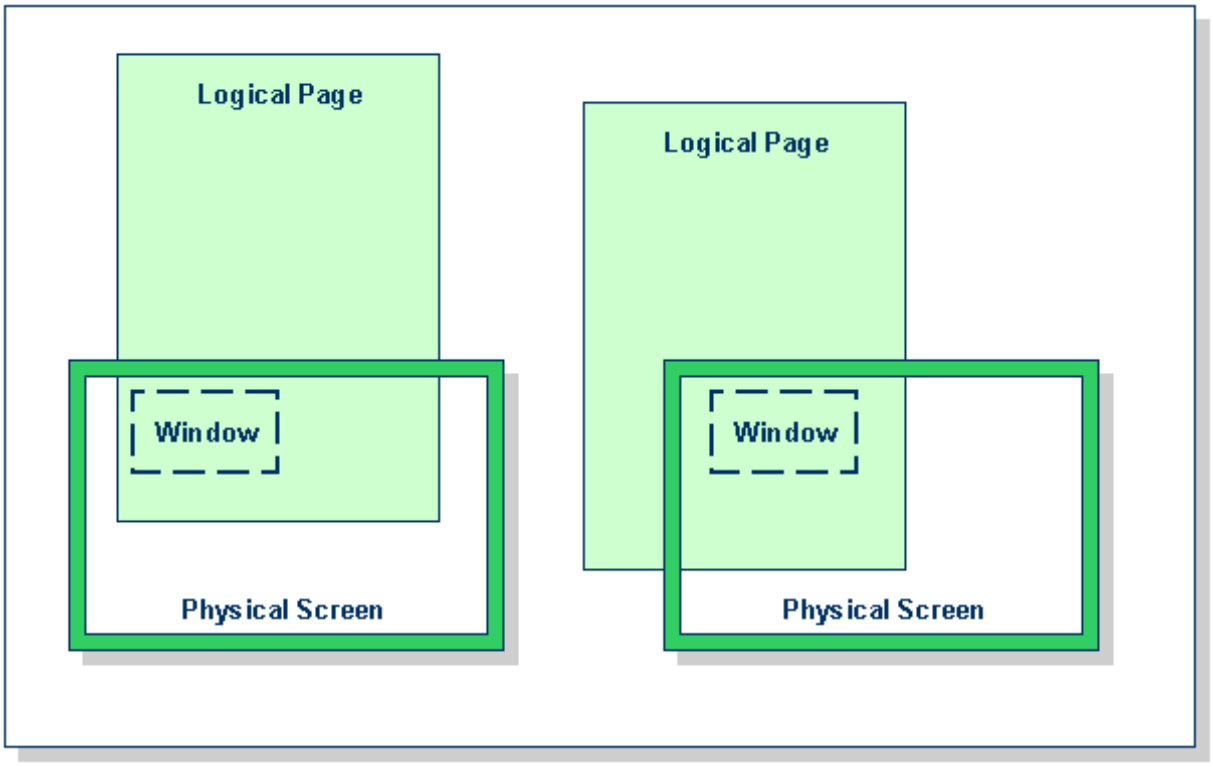
Any incorrect/impossible size or positioning commands will either be ignored or adjusted to the physical possibilities.

Command	Function
%WF	Switch on framing. The boundaries of the window will be indicated by a frame. If the window size is smaller than 4 lines by 12 (or 13 on mainframe computers) columns, the frame will not be visible.
%WM	Switch off framing. The boundaries of the window will not be indicated by a frame. Switching off the frame does not change the size of the window (only the size of the page segment visible inside the window).
%WO	Suppress display of PF-key lines, message line and statistics line. This command only applies if the screen is a "real" window (that is, smaller than the physical screen). To cancel the effect of %WO, you issue %WO again (or %WD).
%WP	By default, the PF-key lines, the message line and the statistics line are displayed within a window. To display them on the screen outside the window, use %WP. To cancel the effect of %WP, you use %WD.
%WD	Cancels the effects of %WF, %WO and %WP (and of the TITLE option of the DEFINE WINDOW statement).
%WX	If there is a *COM field outside the window, this field is normally not write-protected. With %WX you can make it write-protected. This option is only available on mainframe computers.
%WY	Cancels the effect of %WX. This option is only available on mainframe computers.

Window Position on a Logical Page

The following window commands control the positioning of the window on the current logical page, that is, the current report/map produced by the Natural program for display. This logical page may be larger in size than the physical screen.

When you change the position of the window on the logical page, the size and position of the window on the physical screen will remain unchanged. In other words, the window is not moved over the page, but the page is moved "underneath" the window:



Unless specified differently by one of the following commands, the window will be placed at the top left corner of the logical page.

Command	Function
%W*	The position marked on the page by the cursor will be shifted to the top left corner of the window.
%W*lll,ccc*	The position of the logical page determined by line number *lll*/column number *ccc* will be shifted to the top left corner of the window. Lines and columns are counted on the logical page.
%W<	Shift window left. The number of positions shifted is equal to the line size (horizontal extension) of the window.
%W<<	Shift window to leftmost position of page.
%W<*n*	Shift window left *n* positions ($0 \leq n \leq$ logical line size).
%W>	Shift window right. The number of positions shifted is equal to the line size (horizontal extension) of the window.
%W>>	Shift window to rightmost position of page.
%W>*n*	Shift window right *n* positions ($0 \leq n \leq$ logical line size).
%W+	Shift window down. The number of lines shifted is equal to the number of lines in the window. (*)
%W++	Shift window to bottom of page. (*)
%W+*n*	Shift window down *n* lines ($0 \leq n \leq$ logical page size). (*)
%W-	Shift window up. The number of lines shifted is equal to the number of lines in the window.
%W--	Shift window to top of page.
%W-*n*	Shift window up *n* lines ($0 \leq n \leq$ logical page size).
%WH	By default, the position of the window on the logical page is reset to "top left corner" after a screen I/O. %WH prevents the window position from being reset by the next I/O, that is, the set window position will be retained. %WH only applies to the next I/O.
%WS	Switch on the STAY option; that is, control will "stay" on the current page until the end of the page. If a page is not yet completely shown in vertical direction, the indicator "VVVV" will appear in the message line. The window will be scrolled downward with every ENTER until the end of the logical page is reached. The next ENTER will cause control to be returned to the program. (This does not apply to pages created by an INPUT statement with input fields (session parameter AD=A or AD=M).)
%WN	Switch off the STAY option. When you press ENTER, control is returned to the program.

* The window can be shifted to the last non-blank line of the page at most.

Notes:

- If you wish to use one of the above commands within a program to shift the window, assign the command to a function key (with a SET KEY statement).
- If you wish to specify it with a SET CONTROL statement, this statement must be followed by a REINPUT statement (that is, it must be placed between the REINPUT statement and corresponding INPUT statement); otherwise Natural will not be able to uniquely identify the window to which the command is to be applied (and will ignore it).
- As a rule, however, no SET CONTROL 'W' statement should be placed between an INPUT statement with WINDOW='window-name' option and the corresponding REINPUT statement.

Examples of Command Combinations

The various parameters to be specified with the %W command may also be combined with one another; for example:

`%W<<--`	Position window to top left corner of page.
`%W>>++`	Position window to bottom right corner of page.
`%W++-`	Display the next to last window segment of the page.
`%W+3>6`	Position window 3 lines down and 6 positions right on the page.
`%W10+>`	Position window to line 10 of the page, then 1 window down and 1 window right.
`%WL40C10++-3`	Define the window with a line size of 40 positions and a page size of 10 lines, move that window to the bottom of the page, then move it up 3 lines on the page.
`%WL30C10B3/15--<<`	Define the window with a line size of 30 positions, page size of 10 lines, position the window at line 3 column 15 on the physical screen and move this window to the top left corner of the page.
`%WFS`	Generate a frame around the window and set the STAY option on.

Notes:

- When you specify multiple parameters with the %W command, note that on mainframe computers the maximum number of characters after the "%" is 24; any further characters will be ignored.
- The parameters are evaluated in the sequence in which they are specified, so that different sequences of the same parameters may lead to different results.

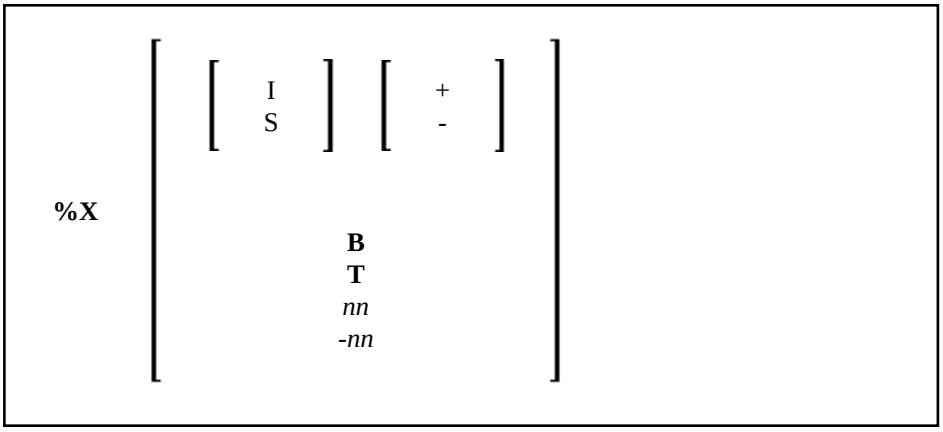
%WA and %WZ - Save Screen Image before Window

Command	Function
%WA	With "%WA", you activate a "save screen image before window" feature. When this feature is activated and a window is to be opened, all active screen data that will be overlaid by the window are saved. When the same window is moved, the saved screen image is reconstructed before the window is built up at the new location on the screen. In addition, it is possible to rebuild the saved images of multiple dependent windows whenever the calling window becomes active again. When the current INPUT statement uses a window, the screen image is stored before the window is output. Whenever the same INPUT statement is repeated, the current or all subsequent stored screen images are recovered and are written back to the screen. This feature makes it possible, for example, to use windows in a PC-like manner. For a given window, any number of dependent windows can be written to the screen. All these windows will disappear from the screen when the main input window is re-executed. The buffer contents (screen images) are deleted whenever Natural performs a full-screen I/O, when Natural returns to command mode (NEXT), after a LOGON command, or after the CLEAR key has been pressed.
%WZ	With "%WZ", you deactivate a previously entered "%WA" command.

%X - Control of Statistics Line/Infoline

Screen Design (Statistics Line/Infoline - Terminal Command %X) in the Natural Programming Guide.

Command Syntax



This terminal command controls the display of the Natural statistics line or infoline.

Natural for Mainframes:
The line can be used as either statistics line or infoline (but not both at the same time). See also Statistics Line/Infoline - Terminal Command %X in the Natural Programming Guide => Designing User Interfaces => Screen Design.

Natural for UNIX, Natural for Windows:
The line can only be used as infoline.

Command	Function
%X+	switches the display of the statistics line/infoline on.
%X-	switches the display of the statistics line/infoline off.
%X	switches the display of the statistics line/infoline on and off (toggle switch).
%XI+	displays the line in infoline mode.
%XI-	displays the line in statistics display mode (only possible on mainframe computers).
%XI	switches between the two display modes (only possible on mainframe computers).
%XS+	displays additional statistics (see below).
%XS-	returns to display of original statistics.
%XS	switches between additional and original statistics.
%XB	displays the statistics line/infoline in the bottom line of the screen.
%XT	displays the statistics line/infoline in the top line of the screen.
%X*nn*	displays the statistics line/infoline on line *nn* of the screen. If the line number *nn* is not within the current screen, the statistics line/infoline will not be displayed.
%X-*nn*	displays the statistics line/infoline on the *nn*th line from the bottom of the screen. If the line number *-nn* is not within the current screen, the statistics line/infoline will not be displayed.

Infoline

Data can be written to the infoline by specifying the output destination "INFOLINE" with the DEFINE PRINTER statement. Only a single line can be written to the infoline. The infoline can be used to have status information displayed, for example, for debugging purposes; it can also be used as separator line (as defined by SAA standards).

Statistics Line - on Mainframe Computers only

When the line is used in statistics display mode, the following statistical information is provided:

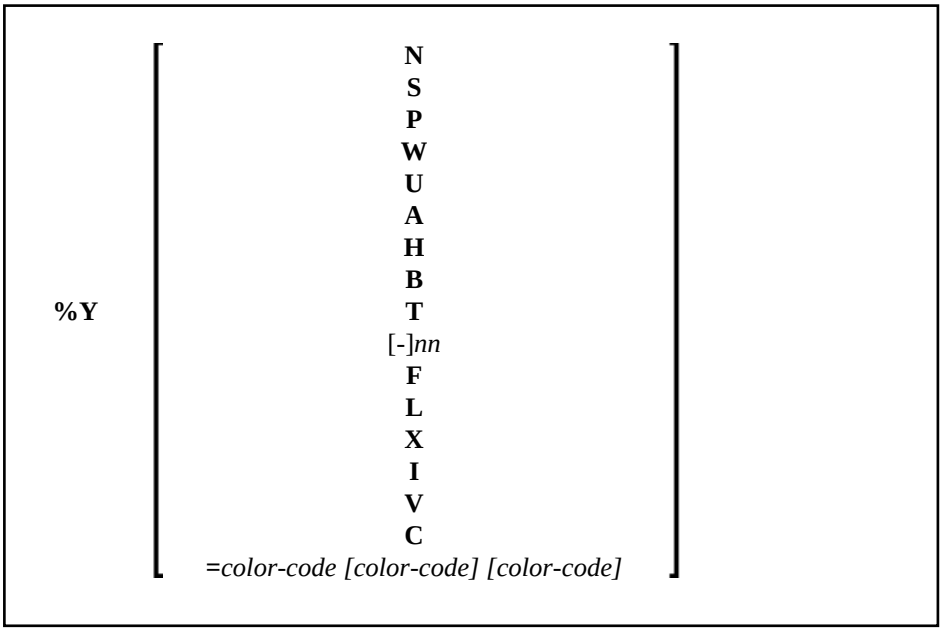
Command Option	Function
IO	Number of bytes transmitted to the screen during the previous screen operation.
AIO	Average number of bytes transmitted per terminal screen operation (since beginning of Natural session).
L	Logical line number of top line within the logical page shown in the current window.
C	Logical column number of the leftmost column within the logical page shown in the current window.
LS	Logical line size of current page (as defined by session parameters).
PS	Logical page size of current page (as defined by session parameters).
PLS	Physical line size of window.
PCS	Physical page size of window.
FLD	Number of fields generated in the last screen.
CLS	Number of program calls during the last terminal I/O.
ADA	Number of Adabas calls during the last terminal I/O.

Additional statistics displayed with %XS+

Command Option	Function
MIN	Minimum duration (in seconds) of Natural activities between two terminal I/Os in the current Natural session.
MAX	Maximum duration (in seconds) of Natural activities between two terminal I/Os in the current Natural session.
AVR	Average duration (in seconds) of Natural activities between two terminal I/Os in the current Natural session.
LST	Duration (in seconds) of Natural activities between the last but one and the last but two terminal I/Os.

%Y - Control of PF-Key Lines

Command Syntax



The terminal command %Y is used to control the display of the Natural PF-key lines.

Note:
On graphical user interfaces, this command is ignored.

The following topics are covered below:

- Display Format of Function-Key Lines
- Single- and Double-Line Display
- Positioning of Function-Key Lines
- Range of Displayed Function Keys
- Intensified or Reverse Video Display of Function-Key Lines
- Coloring of Function-Key Lines
- Cursor Sensitivity

Display Format of Function-Key Lines

Command	Function
%YN	displays the function-key lines in normal tabular Software AG format.
%YS	displays the function-key lines in sequential format, and only showing those keys to which names have been assigned (PF1=value,PF2=value,etc.).
%YP	displays the function-key lines in PC-like sequential format; corresponds to %YS, except that "Fn=" instead of PFn=" is displayed before the names.
%YW	corresponds to %YP - but applies only if the function-key lines are displayed within a window. This option is only available on mainframe computers.
%YU	cancels the effect of %YW. This option is only available on mainframe computers.

Single- and Double-Line Display

Command	Function
%YA	(all-line display) displays both function-key lines.
%YH	(half display) displays only one function-key line; for normal tabular display mode (%YN), this is the line with the function-key names; for the other two display modes (%YS and %YP), this is the upper line to be displayed.

Positioning of Function-Key Lines

Command	Function
%YB	displays the function-key lines at the bottom of the screen.
%YT	displays the function-key lines at the top of the screen.
%Y*nn*	displays the function-key lines on line *nn* on the screen.
%Y-*nn*	displays the function-key lines on the *nn*th line from the bottom of the screen. If the line number *nn* or *-nn* is not within the current screen, the function-key lines will not be displayed.

Range of Displayed Function Keys

Command	Function
%YF	displays the first range of function keys (that is, usually 1 to 12).
%YL	displays the last range of function keys (that is, usually 13 to 24).
%YX	is used to toggle-switch between the two displays.

Intensified or Reverse Video Display of Function-Key Lines

Command	Function
%YI	displays the function-key lines intensified. Enter "%YI" again to switch back from intensified to non-intensified display.
%YV	displays the function-key lines in reverse video. Enter "%YV" again to switch back from reverse video display to normal display.

Coloring of Function-Key Lines

Command	Function
%Y=*color-code(s)*	displays the function-key lines in the specified colors. The *color-codes* you can specify are the same as with the session parameter CD). You can specify up to three color-codes: the first *color-code* refers to the first function-key line (the one displaying the function-key numbers), the second color-code refers to the second function-key line (the one displaying the function-key names), and the third *color-code* refers to the background of both lines. For example, %Y=GRPIYE would cause the characters in the first line to be displayed in green, the characters in the second line in pink, and the background of both lines in yellow. **Note**: On mainframe computers, the third "color-code" will be ignored.

Cursor Sensitivity

Command	Function
%YC	This command makes the function-key lines cursor-sensitive. This means that they react like an action bar on a PC screen: the user just moves the cursor to the desired function-key number or name displayed and presses ENTER, and Natural will react as if the corresponding function key had been pressed. Enter "%YC" again to switch cursor-sensitivity off again (toggle switch). By using %YC in conjunction with tabular Software AG display format (%YN) and having only the function-key names displayed (%YH), you may equip your applications with very comfortable action bar processing: the user merely has to select a function name with the cursor and press ENTER, and the function will be executed.

To activate the display of the function-key lines in a program, you use the session parameter KD=ON.

%Z - Clear Source Area

Command Syntax

%Z

This command clears the contents of the Natural source area.

Note:
This command can only be used with the SET CONTROL statement.

